Final Report for AOARD Grant FA2386-13-1-4043

"Non-Metric Similarity Measures"

Date: March 26, 2015

Name of Principal Investigator: Kai Ming Ting

E-mail: kaiming.ting@federation.edu.au

Institution: Federation University

Mailing Address: PO Box 3191 Gippsland Mail Centre VIC 3841, Australia

Phone: +61 3 512 26241

Period of Performance: 3/28/2013 - 3/27/2015

Abstract

All three project aims have been achieved by the end of the project period, i.e., create two non-metric similarity measures and three relative mass measures, elicit the relative functions of unary measures and binary measures, and evaluate the new measures in four data mining tasks—two additional to the two tasks specified in the project proposal. The non-metric similarity measures were created as a generalisation of mass estimation from a unary function to a binary function. A derivative of mass measure called relative mass was also investigated using three implementations. The research in relative mass was expanded (outside the project scope) to two tasks: In anomaly detection, relative mass is used to overcome one weakness of current mass-based anomaly detectors using a tree-based approach and a nearest-neighbour-based approach; in clustering, relative mass is used to recondition density-based clustering algorithms to successfully find clusters with varying densities.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **26 MAR 2015**	2. REPORT TYPE **Final**	3. DATES COVERED **28-03-2013 to 27-03-2015**

4. TITLE AND SUBTITLE
Non-Metric Similarity Measures

5a. CONTRACT NUMBER
FA2386-13-1-4043

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)
Kai Ming Ting

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Federation University,PO Box 3191,Gippsland Mail Centre,VIC 3841 Australia,NA,NA

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AOARD, UNIT 45002, APO, AP, 96338-5002

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/AFOSR/IOA(AOARD)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AOARD-134043

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This was an extension of mass theory to non-metric similarity measures. The non-metric similarity measures were created as a generalization of mass estimation from a unary function to a binary function. Unlike the traditional similarity measure that is based on geometric difference between two instances, the mass-based measure takes data distribution into account. It is demonstrated that the new measure results in better performance in applying to information retrieval task. A derivative of mass measure called relative mass was also investigated using three implementations. The research in relative mass was expanded to two tasks: In anomaly detection relative mass is used to overcome one weakness of current mass-based anomaly detectors using a tree-based approach and a nearest-neighbor-based approach; in clustering, relative mass is used to recondition density-based clustering algorithms to successfully find clusters with varying densities.

15. SUBJECT TERMS
Mass theory, Density estimation, Similarity measure, Non-metric measure, Nearest neighbor, Density-based clustering, Anomaly detection, Information retrieval, Classification

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**27**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

These works have been reported in five papers, where three have been published, one technical report and one paper is currently under review. In addition, the works on building mass-based methods using a nearest neighbour approach and its extension to apply to Bayesian classifier learning, supported by a previous AOARD project, have been published in Pattern Recognition Journal and Computational Intelligence Journal.

1 Introduction

Two previous projects, supported AOARD from 2010 to 2013, have pioneered mass estimation and shown that it is an effective and efficient alternative to density estimation in handling five data mining tasks: information retrieval, regression, anomaly detection, clustering and Bayesian classification. This project deepens the impact already achieved using mass estimation to elicit the utility of non-metric similarity measures in data mining tasks.

This project aims to

1. **Create non-metric similarity measures for numeric data.**
2. **Elicit the relative functions of unary measures, as currently established in mass estimation, and binary similarity measures in solving data mining problems.**
3. **Evaluate the new measures in classification and information retrieval tasks.**

The ultimate goal of the work is to find answers to the following two fundamental research questions:

1. To compute similarity between any two instances, do we have to use a metric?
2. Do we have to compute similarity/distance between instances to solve a data mining problem?

All three project aims have been achieved by the end of the project period. This report provides the findings in a more concise form, extracted from the papers [3, 4, 5, 6, 7] produced from this research. The theoretical analyses are provided in Section 2, the results and discussion in Section 3, and the final remark in Section 4.

2 Theoretical Analyses

This section describes the theoretical analyses of non-metric similarity measures and a derivative of mass measure called relative mass in the following two subsections.

2.1 m_p dissimilarity measure

The new dissimilarity measure uses data distribution as the primary contributor in measuring dissimilarity between instances. Rather than using a spatial distance in each dimension, m_p-dissimilarity evaluates the dissimilarity between two instances in terms of probability mass in a region covering the two instances in each dimension. The final dissimilarity between the two instances is estimated as a power mean of dissimilarities in each dimension as in ℓ_p-norm. The intuition behind the proposed dissimilarity measure is that two instances are likely to be more dissimilar if there are more instances in between and around them in many dimensions. Under the proposed data dependent dissimilarity measure, two instances in a dense region of the distribution are more dissimilar than two instances having the same geometric distance in a sparse region, as prescribed by psychologists.

In order to measure dissimilarity between $\mathbf{x}$ and $\mathbf{y}$, instead of using $(x_i - y_i)$ in ℓ_p-norm, we propose to consider the relative positions of $\mathbf{x}$ and $\mathbf{y}$ with respect to the rest of the data distribution in each dimension. The dissimilarity between $\mathbf{x}$ and $\mathbf{y}$ in dimension i can be estimated as the probability data mass in a region $R_i(\mathbf{x}, \mathbf{y})$ that encloses $\mathbf{x}$ and $\mathbf{y}$. If there are many instances in $R_i(\mathbf{x}, \mathbf{y})$, $\mathbf{x}$ and $\mathbf{y}$ are likely to be more dissimilar in dimension i. Using the same power mean formulation as in ℓ_p-norm, the data dependent dissimilarity measure based on probability mass can be defined as:

$$m_p(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{d} \left(\frac{|R_i(\mathbf{x}, \mathbf{y})|}{n} \right)^p \right)^{\frac{1}{p}} \tag{1}$$

where $|R_i(\mathbf{x}, \mathbf{y})|$ is the data mass in region $R_i(\mathbf{x}, \mathbf{y})$, $R_i(\mathbf{x}, \mathbf{y}) = [\min(x_i, y_i) - \delta_i, \max(x_i, y_i) + \delta_i]$, $\delta_i \geq 0$ and n is the number of data instances. An example of $R_i(\mathbf{x}, \mathbf{y})$ is shown in Figure

1. We use $\delta_i = \frac{\sigma_i}{2}$ (σ_i is the standard deviation of data in dimension i) in this paper.

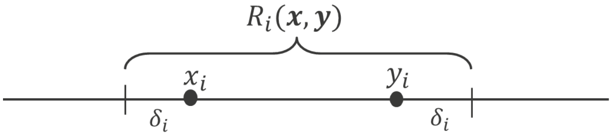


Figure 1: $R_i(\mathbf{x}, \mathbf{y})$

We call the proposed dissimilarity measure $m_p(\mathbf{x}, \mathbf{y})$ 'm_p-dissimilarity'. This measure captures the essence of the distance-density model proposed by psychologists which prescribes that two instances in a sparse region are more similar than two instances in a dense region. Although m_p employs the same power mean formulation as ℓ_p, the core calculation is based on mass rather than distance. It signifies the degree of dissimilarity: the higher the measure, the more dissimilar the two instances are; just like ℓ_p.

The formulation of $m_p(\mathbf{x}, \mathbf{y})$ (Eqn. 1) has a probabilistic interpretation (we refer the reader to the attached paper for details).

2.2 Relative Mass

A derivative of mass called relative mass is introduced to overcome one weakness of the basic (unary) mass measure. As a global measure, mass has been shown to be an efficient and effective alternative to density in modelling data distribution to solve different data mining problems [11]. However, some problems require a local measure which takes local distribution into consideration. For example, in the anomaly detection context, density-based anomaly detectors has been shown to have difficulty detecting local anomalies if the basic density is employed. A relative density measure such as Local Outlier Factor [15] has been proposed to overcome this weakness. Relative mass follows the same idea. Indeed, it overcomes the same issue in mass-based anomaly detector such as iForest [8].

Two implementations of relative mass have been created. The first is based on iForest [8] and the second is based nearest neighbour implementation of mass estimation [1]. These are described in the following two subsections.

2.2.1 ReMass-iForest

This section describes iForest and its weakness in detecting local anomalies and introduces the new anomaly detector, ReMass-iForest, based on the relative mass to overcome the weakness.

iForest

Given a d-variate database of n instances ($D = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(n)}\}$), iForest [8] constructs t iTrees ($T_1, T_2, \cdots, T_t$). Each T_i is constructed from a small random sub-sample ($\mathcal{D}_i \subset D$, $|\mathcal{D}_i| = \psi < n$) by recursively dividing it into two non-empty nodes through a randomly selected attribute and split point. A branch stops splitting when the height reaches the maximum (H_{max}) or the number of instances in the node is less than $MinPts$. The default values used in iForest are $H_{max} = \log_2(\psi)$ and $MinPts = 1$. The anomaly score is estimated as the average path length over t iTrees as follows:

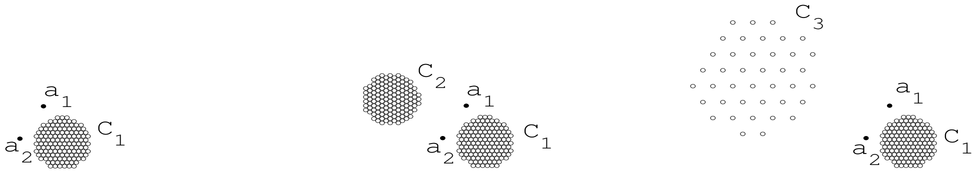
$$L(\mathbf{x}) = \frac{1}{t} \sum_{i=1}^{t} \ell_i(\mathbf{x}) \tag{2}$$

where $\ell_i(\mathbf{x})$ is the path length of $\mathbf{x}$ in T_i
As anomalies are likely to be isolated early, they have shorter average path lengths. Once all instances in the given data set have been scored, the instances are sorted in ascending order of their scores. The instances at the top of the list are reported as anomalies.
iForest runs very fast because it does not require distance calculation and each iTree is constructed from a small random sub-sample of data.
iForest is effective in detecting global anomalies (e.g., a_1 and a_2 in Figures 2a and 2b) because they are more susceptible to isolation in iTrees. But it fails to detect local anomalies (e.g., a_1 and a_2 in Figure 2c) as they are less susceptible to isolation in iTrees. This is because the local anomalies and the normal cluster C_3 have about the same density. Some fringe instances in the normal cluster C_3 will have shorter average path lengths than those for a_1 and a_2.

ReMass-iForest

In each iTree T_i, the anomaly score of an instance $\mathbf{x}$ w.r.t its local neighbourhood, $s_i(\mathbf{x})$, can be estimated as the ratio of data mass as follows:

$$s_i(\mathbf{x}) = \frac{m(\breve{T}_i(\mathbf{x}))}{m(T_i(\mathbf{x})) \times \psi} \tag{3}$$



(a) Global anomalies (b) Global anomalies (c) Local anomalies

Figure 2: Global and Local anomalies. Note that both anomalies a_1 and a_2 are exactly the same instances in Figures (a), (b) and (c). In Fig.(a) and Fig.(b), a_1 and a_2 have low density than that in the normal clusters C_1 and C_2. In Fig.(c), a_1, a_2 and the normal cluster C_3 have the same density but a_1 and a_2 are anomalies relative to the normal cluster C_1 with a higher density.

where $T_i(\mathbf{x})$ is the leaf node in T_i in which $\mathbf{x}$ falls, $\breve{T}_i(\mathbf{x})$ is the immediate parent of $T_i(\mathbf{x})$, and $m(\cdot)$ is the data mass of a tree node. ψ is a normalisation term which is the training data size used to generate T_i.

$s_i(\cdot)$ is in $(0, 1]$ because a parent node has mass values ranging from 2 to ψ in an iTree created from a training set of ψ instances. The higher the score the higher the likelihood of $\mathbf{x}$ being an anomaly. Unlike $\ell_i(\mathbf{x})$ in iForest, $s_i(\mathbf{x})$ measures the degree of anomaly locally.

The final anomaly score can be estimated as the average of local anomaly scores over t iTrees as follows:

$$S(\mathbf{x}) = \frac{1}{t}\sum_{i=1}^{t} s_i(\mathbf{x}) \tag{4}$$

Once every instance in the given data set has been scored, instances can be ranked in descending order of their anomaly scores. The instances at the top of the list are reported as anomalies.

Relation to LOF and DEMass-LOF

The idea of relative mass in ReMass-iForest has some relation to the idea of relative density in Local Outlier Factor (LOF) [15]. LOF uses k nearest neighbours to estimate density $\bar{f}_k(\mathbf{x}) = \dfrac{|N(\mathbf{x}, k)|}{n \sum_{\mathbf{x}' \in N(\mathbf{x},k)} distance(\mathbf{x}, \mathbf{x}')}$ where $N(\mathbf{x}, k)$ is the set of k nearest neighbours of $\mathbf{x}$. It estimates its anomaly score as the ratio of the average density of $\mathbf{x}$'s k nearest neighbours to $\bar{f}_k(\mathbf{x})$. In LOF, the local neighbourhood is defined by k nearest neighbours which requires distance calculation. In contrast, in ReMass-iForest, the local neighbourhood is

Table 1: Ranking measure and complexities (time and space) of ReMass-iForest, iForest, DEMass-LOF and LOF.

	ReMass-iForest	iForest	DEMass-LOF	LOF
Ranking Measure	$\frac{1}{t\psi}\sum_{i=1}^{t}\frac{m(\breve{T}_i(\mathbf{x}))}{m(T_i(\mathbf{x}))}$	$\frac{1}{t}\sum_{i=1}^{t}\ell_i(\mathbf{x})$	$\frac{\sum_{i=1}^{t}\frac{m(\breve{T}_i(\mathbf{x}))}{\breve{v}_i}}{\sum_{i=1}^{t}\frac{m(T_i(\mathbf{x}))}{v_i}}$	$\frac{\sum_{\mathbf{x}'\in N(\mathbf{x},k)}\frac{\bar{f}_k(\mathbf{x}')}{\lvert N(\mathbf{x},k)\rvert}}{\bar{f}_k(\mathbf{x})}$
Time Complexity	$O(t(n+\psi)\log\psi)$	$O(t(n+\psi)\log\psi)$	$O(t(n+\psi)bd)$	$O(dn^2)$
Space Complexity	$O(t\psi)$	$O(t\psi)$	$O(td\psi)$	$O(dn)$

$\breve{v}_i$ and v_i are the volumes of nodes $\breve{T}_i(\mathbf{x})$ and $T_i(\mathbf{x})$, respectively.

the immediate parent in iTrees. It does not require distance calculation.

DEMass-LOF [9] computes the same anomaly score as LOF from trees, without distance calculation. The idea of relative density of parent and leaf nodes was used in DEMass-LOF. It constructs a forest of t balanced binary trees where the height of each tree is $b \times d$ (b is a parameter that determines the level of division on each attribute and d is the number of attributes). It estimates its anomaly score as the ratio of average density of the parent node to the average density of the leaf node where $\mathbf{x}$ falls. The density of a node is estimated as the ratio of mass to volume. It uses mass to estimate density and ranks instances based on the density ratio. Like iForest, it is fast because no distance calculation is involved. But, it has limitation in dealing problems with even a moderate number of dimensions because each tree has $2^{(b\times d)}$ leaf nodes.

In contrast to LOF and DEMass-LOF, ReMass-iForest does not require density estimation, it uses relative mass directly in order to estimate the local anomaly score from each iTree.

The ranking measure and complexities (time and space) of ReMass-iForest, iForest, DEMass-LOF and LOF are provided in Table 1.

2.2.2 iNNE

The intuition of iNNE comes from the fact that an anomaly is expected to be far from its nearest neighbour; and the reverse is true for a normal instance. Thus, we propose to use a small sample from the given data set and build a region around each instance in order to isolate it from the rest of the instances. The instance to be isolated is placed at the centre of the region and the boundary of the region is defined by the distance to the instance's nearest neighbour. The sample size determines the number of regions to be created. A sample size of ψ will produce ψ regions in order to isolate each and every instance in the sample. Because of the use of nearest neighbour to determine the boundary of a region, the size of the region adapts to the data distribution: large regions in sparse area and small regions in dense area.

Like iForest, iNNE isolates each instance in a subsample and builds an ensemble from multiple subsamples. We formally define iNNE as follows.

Let $\mathcal{S} \subset D$ be a subsample of size ψ selected randomly without replacement from a dataset $D \subset \Re^d$, and let $\|\mathbf{x}-\mathbf{y}\|$ denote the Euclidean distance between instances $\mathbf{x}$ and $\mathbf{y}$, where $\mathbf{x}, \mathbf{y} \in \Re^d$.

η_c is the nearest neighbour of c, and $\tau(c) = \|c - \eta_c\|$, where $c,\ \eta_c \in \mathcal{S}$

$\mathbb{B}(c)$, a hypersphere centred at c with radius $\tau(c)$, is defined to be $\{\mathbf{x} : \|\mathbf{x} - c\| < \tau(c)\}$

Note that $\mathbb{B}(c)$ is the largest hypersphere which isolates instance c from the rest of the instances in $\mathcal{S}$. Its radius $\tau(c)$ is a measure of the degree of isolation of c. The larger the radius, the more isolated c is; and vice versa. Also the relative size of $\mathbb{B}(c)$ and $\mathbb{B}(\eta_c)$ is a measure of isolation of c relative to its neighbourhood. Such a measure is defined below.

Isolation score $I(\mathbf{x})$ based on $\mathcal{S}$ is defined as follows:

$$I(\mathbf{x}) = \begin{cases} 1 - \dfrac{\tau(\eta_c)}{\tau(c)} & \text{if } \mathbf{x} \in \bigcup\limits_{c \in \mathcal{S}} \mathbb{B}(c) \\ \\ 1 \quad \text{otherwise} \end{cases}$$

where $\tau(c) = min\{\tau(d) : \mathbf{x} \in \mathbb{B}(d),\ d \in \mathcal{S}\}$

From the above definitions, we can deduce that $0 \leq I(\mathbf{x}) \leq 1$, because $\dfrac{\tau(\eta_c)}{\tau(c)} \leq 1$.

iNNE has an ensemble of hyperspheres $\{\bigcup\limits_{c \in \mathcal{S}_i} \mathbb{B}(c) \mid i = 1, \ldots, t\}$, generated from t subsamples $S_i,\ \ i = 1, \ldots, t$.

The anomaly score based on iNNE is defined as follows: For every $\mathbf{x} \in \Re^d$,

$$\bar{I}(\mathbf{x}) = \frac{1}{t} \sum_{i=1}^{t} I_i(\mathbf{x})$$

where $I_i(x)$ is the isolation score based on $\mathcal{S}_i$

Based on the anomaly score defined above, instances are ranked in descending order and the highest ranked instances are more likely to be anomalies.

iNNE is implemented as a two-stage process: (i) In training stage, t models as defined in Definition 2.2.2 are built from t randomly selected subsamples of sample size ψ. (ii) In evaluation stage, each test instance is evaluated against every model in iNNE and the isolation scores from t models are averaged to produce the anomaly score as defined in Definition 2.2.2.
In training stage, nearest neighbour search is required in building each of the t models, which accounts for time complexity of $O(t\psi^2)$ and space complexity of $O(t\psi)$. In the second stage, distance is calculated between n instances and every training instance in a model. Since this is done for t models, it accounts for time complexity of $O(nt\psi)$. Thus, the time complexity is dominated by that in the evaluation stage and is linear with respect to n.

Comparing iNNE, iForest and LOF

iNNE, being an isolation based anomaly detection approach, inherits the concept of isolation from iForest. The formulation of the isolation score in iNNE is influenced by the relative density score used in LOF. Table 2 provides a concise comparison between iNNE, iForest and LOF.

Note that the degree of isolation used in both iForest and iNNE is a proxy to mass. In the case of iForest, a region of high mass is expected to have a high number of partitions to isolate an instance in the region. In the case of iNNE, a region of high mass is expected to have a small radius hypersphere to isolate an instance in the region. Thus, the anomaly score used by iNNE is viewed to be a variant of relative mass.

	iNNE	**iForest**	**LOF**
Key mechanism	Isolation	Isolation	Density
Training set	Randomly selected sample from the dataset	Randomly selected sample from the dataset	Entire dataset
Model	Space partitioning using hypersphere	Axis parallel space partitioning	No explicit model
Anomaly score	$1 - \frac{\tau(\eta_c)}{\tau(c)}$	Number of axis parallel partitions required to isolate	Ratio of the density of $\mathbf{x}$'s local neighbourhood and the density of $\mathbf{x}$.
Dimensions used	All dimensions	Subset of dimensions	All dimensions
Time	$O(nt\psi)$	$O(nt\psi)$	$O(n^2)$
Space	$O(t\psi)$	$O(t\psi)$	$O(n)$

Table 2: Comparison between iNNE, iForest and LOF in terms of base concept, methodology and complexity

3 Results and Discussion

This section provides the results and discussion for non-metric similarity measures and relative mass in the following two subsections.

3.1 m_p-dissimilarity measure

We evaluated the performance of m_p against ℓ_p and cosine distance in kNN classification and information retrieval. Eleven data sets from different domains with different sizes ($1000 \leq n \leq 9100$), number of dimensions ($188 \leq d \leq 10000$) and number of classes ($2 \leq c \leq 52$) were used. All the attributes in the data sets are numeric. Out of 11 data sets used, six are from text mining domain, two from music classification and retrieval domain, 2 from character recognition and the last one is a synthetic data set from UCI machine learning repository. Text data were represented by TFIDF weighted 'bag of words' vectors. Other data sets (non-text) were normalised to the range of [0,1].

All classification experiments were conducted using a 10-fold cross validation. We used four settings of p $(2.0, 1.0, 0.5, 0.1)$ in ℓ_p and m_p and two settings of k ($k = 1$ and $k = 10$) for all classifiers. The average accuracy (%) over a 10-fold cross validation is reported. The accuracies of two algorithms are considered to be significantly different if their confidence intervals (based on ± one standard error) do not overlap. The best average classification accuracy over a 10-fold cross validation achieved by m_p, ℓ_p and cosine distance in all 11 data sets is presented in Figure 3. A red dot on the top of the bar indicates that the best performer had significantly better classification accuracy than the other two contenders.

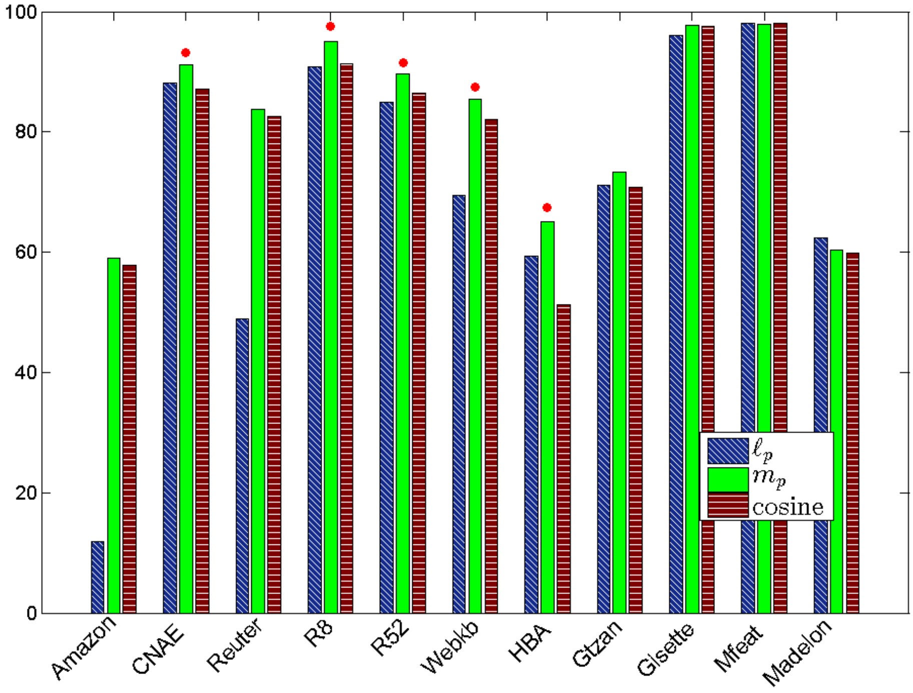


Figure 3: The best classification accuracies of ℓ_p, m_p and cosine distance in kNN classifier. A red dot on the top signifies that the best performer had significantly better classification accuracy than the other two contenders.

As shown in Figure 3, m_p produced better classification accuracies than ℓ_p and cosine distance in eight data sets and similar results in the other three data sets. The result is statistically significant in five data sets (CNAE, R8, R52, Webkb and HBA) and not significantly worst in any data set.

It is interesting to note that m_p produced significantly better classification accuracy than ℓ_p in all six text (sparse) data sets; and better than cosine distance in four out of six. This is because m_p assigns (i) the maximum dissimilarity (of a dimension) if the majority of instances have the same value which is often the case in sparse text data where term frequencies are zeros in many dimensions; and (ii) the minimum dissimilarity if the value has the least number of training instances in the local neighbourhood.

In terms of p, m_p produced better results with $p = 2$ in eight out of 11 data sets used with the exceptions of Amazon ($p = 0.5$), CNAE ($p = 0.1$) and Madelon ($p = 0.1$). The result with ℓ_p, was mixed: $p = 0.1$ produced better classification result in four data sets, $p = 2$ was better in four, $p = 1$ was better in two and 0.5 was better in one data set. Generally, we observed that $p = 2$ is a reasonable setting in m_p, but we can not say anything about setting p in ℓ_p as the accuracy varies significantly with p.

Similar results were observed in the information retrieval tasks. We refer the reader to the attached paper of the detail empirical results.

3.2 Relative Mass

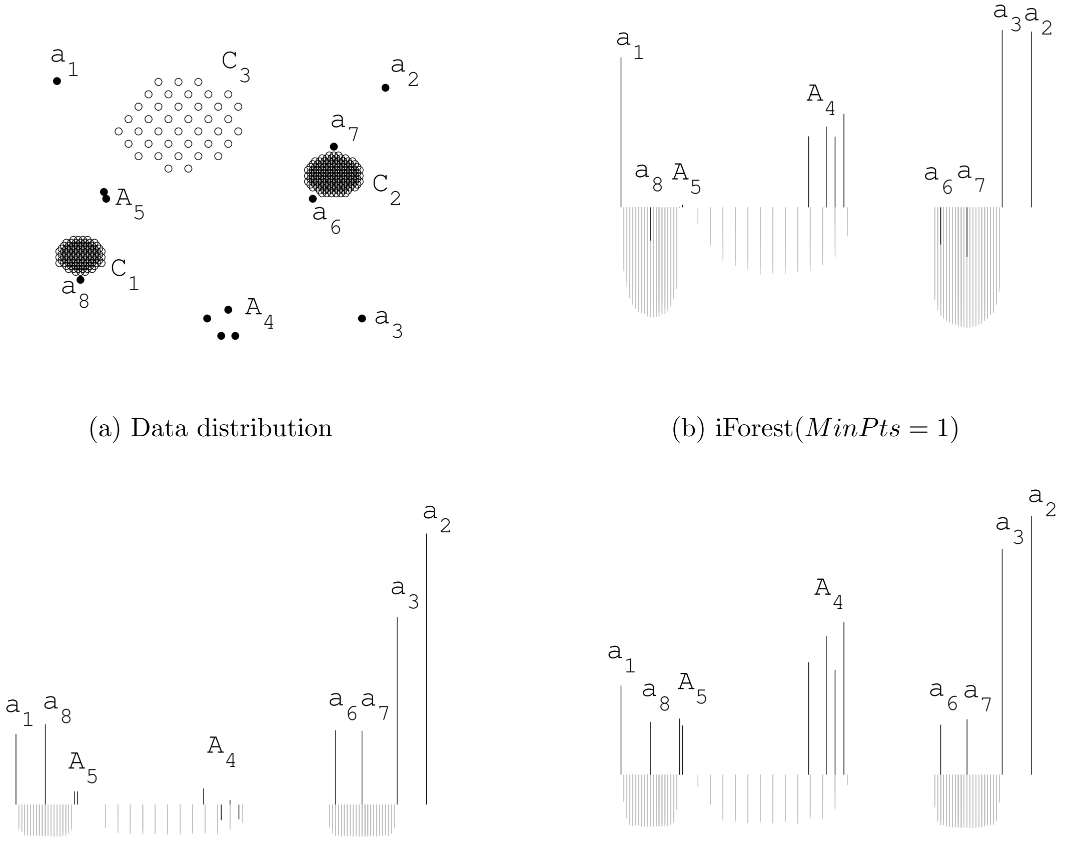
This section presents the results of two implementations of relative mass and applied in anomaly detection. The tree implementation is described in the first subsection and the nearest neighbour implementation in the second subsection.

3.2.1 ReMass-iForest

Two experiments are conducted to compare the anomaly detection accuracy of ReMass-iForest and iForest.

In the first experiment, a synthetic data set is used to demonstrate the ability of ReMass-iForest to detect local anomalies. The data set has 263 normal instances in three clusters and 12 anomalies representing global, local and clustered anomalies. The data distribution is shown in Figure 4a. Instances a_1, a_2 and a_3 are global anomalies; four instances in A_4 and two instances in A_5 are clustered anomalies; and a_6, a_7 and a_8 are local anomalies; C_1, C_2 and C_3 are normal instances in three clusters of varying densities.

Figures 4b-4d show the anomaly scores of all data instances obtained from iForest and ReMass-iForest. With iForest, local anomalies a_6, a_7 and a_8 had lower anomaly scores than some normal instances in C_3; and it produced AUC of 0.98. In contrast, ReMass-iForest had ranked local anomalies a_6, a_7, a_8 higher than any instances in normal clusters C_1, C_2 and C_3 along with global anomalies a_1, a_2 and a_3. But, ReMass-iForest with $MinPts = 1$ did not rank clustered anomalies in A_4 higher than all normal instances, and it produced AUC of 0.99. One fringe instance in the cluster C_3 was ranked higher than two clustered anomalies in A_4. This is because cluster anomalies have similar mass ratio w.r.t their



(a) Data distribution

(b) iForest($MinPts = 1$)

(c) ReMass-iForest($MinPts = 1$)

(d) ReMass-iForest($MinPts = 5$)

Figure 4: Anomaly scores by iForest and ReMass-iForest using $t = 100, \psi = 256$. Note that in anomaly score plots, instances are represented by their values on x_1 dimension. Anomalies are represented by black lines and normal instances are represented by gray lines. The height of lines represents the anomaly scores. In order to differentiate the scores of normal and anomaly instances, the maximum score for normal instances is subtracted from the anomaly scores so that all normal instances have score of zero or less.

Table 3: AUC and runtime (seconds) of ReMass-iForest (RM), iForest (IF), DEMass-LOF (DM), and LOF in benchmark datasets.

			AUC				Runtime			
Data set	n	d	RM	IF	DM	LOF	RM	IF	DM	LOF
Http	567K	3	1.00	1.00	0.99	1.00	71	99	19	19965
ForestCover	286K	10	0.96	0.88	0.87	0.94	42	56	4	2918
Mulcross	262K	4	1.00	1.00	0.99	1.00	20	23	16	2169
Smtp	95K	3	0.88	0.88	0.78	0.95	10	12	16	373
Shuttle	49K	9	1.00	1.00	0.95	0.98	4	9	7	656
Mammography	11K	6	0.86	0.86	0.86	0.68	1	1	5	127
Satellite	6K	36	0.71	0.70	0.55	0.79	1	4	0.6	24
Breastw	683	9	0.99	0.99	0.98	0.96	0.1	0.4	0.3	0.4
Arrhythmia	452	274	0.80	0.81	0.52	0.80	0.3	0.5	5	1
Ionosphere	351	32	0.89	0.85	0.85	0.90	2	3	0.5	0.3

parents as that for the instances in sparse normal cluster C_3. Clustered anomalies were correctly ranked and AUC of 1.0 was achieved when $MinPts$ was increased to 5. The performance of iForest did not improve when $MinPts$ was increased to any values in the range (2, 3, 4, 5 and 10).

In the second experiment, we used the ten benchmark data sets previously employed by Liu et al (2008) [8]. In ReMass-iForest, iForest and DEMass-LOF, the parameter t was set to 100 as default and the best value for the sub-sample size ψ was searched from 8, 16, 32, 64, 128 to 256. In ReMass-iForest, $MinPts$ was set to 5 as default. iForest uses the default settings as specified in [8], i.e, $MinPts = 1$. The level of subdivision (b) for each attribute in DEMass-LOF was searched from 1, 2, 3, 4, 5, and 6. In LOF, the best k was searched between 5 and 4000 (or to $\frac{n}{4}$ for small data sets), with steps from 5, 10, 20, 40, 60, 80, 150, 250, 300, 500, 1000, 2000, 3000 to 4000. The best results were reported. The characteristics of the data sets, AUC and runtime (seconds) of ReMass-iForest, iForest, DEMass-LOF and LOF are presented in Table 3.

In terms of AUC, ReMass-iForest had better or at least similar results to iForest. Based on the two-standard-error significance test, it produced better results than iForest in the ForestCover and Ionosphere data sets. Most of these datasets do not have local anomalies. So, both methods had similar AUC in eight data sets. Note that iForest did not improve AUC when $MinPts$ was set to 5. ReMass-iForest had produced significantly better AUC than DEMass-LOF in relatively high dimensional data sets (Arrhythmia - 274, Satellite

- 36, Ionosphere - 32, ForestCover - 10, Shuttle - 9). These results show that DEMass-LOF has problem in handling data sets with a moderate number of dimensions (9 or 10). ReMass-iForest was competitive to LOF. It was better than LOF in the Mammography data set, worse in the Smtp and Satellite data sets, and equal performance in the other seven data sets.

As shown in Table 3, the runtime of ReMass-iForest, iForest and DEMass-LOF were of the same order of magnitude whereas LOF was upto three order of magnitude slower in large data sets. Note that we can not conduct a head-to-head comparison of runtime of ReMass-iForest and iForest with DEMass-LOF and LOF because they were implemented in different platforms (MATLAB versus JAVA). The results are included here just to provide an idea about the order of magnitude of runtime. The difference in runtime of ReMass-iForest and iForest was due to the difference in ψ and $MinPts$. $MinPts = 5$ results in smaller size iTrees in ReMass-iForest than those in iForest ($MinPts = 1$). Hence, ReMass-iForest runs faster than iForest even though the same ψ is used.

3.2.2 iNNE

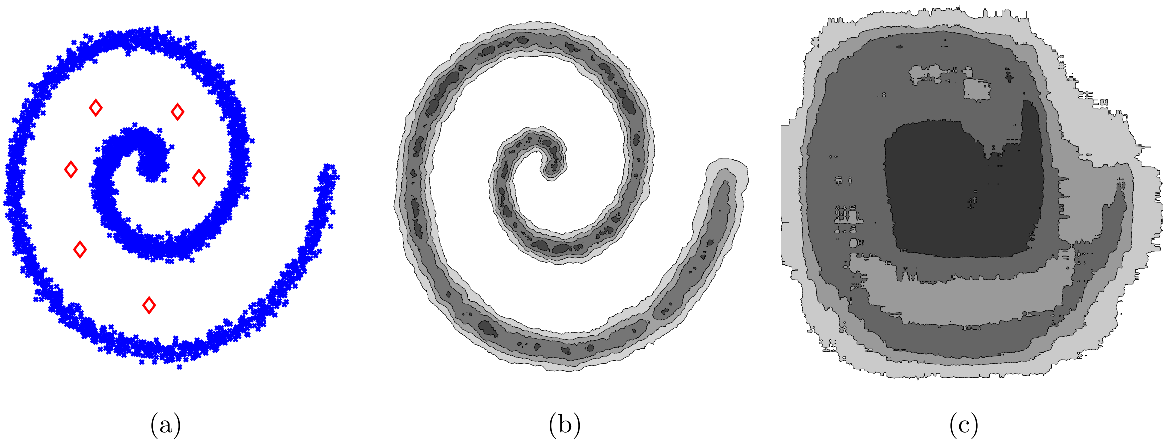
Because of the use of relative mass, iNNE can detect local anomalies as well as ReMass-iForest. This result can be found in [5].

Because of the use of non-axis-parallel partitions, the contour of anomaly score of iNNE is much better than that of iForest. This result is shown in the following subsection.

Anomalies surrounded by normal clusters

When anomalies are surrounded by normal clusters, they are masked by normal instances in axis-parallel projections. Since, iForest uses axis-parallel subdivisions to isolate anomalies, it cannot isolate anomalies which are masked in axis parallel projections. In contrast, iNNE employs non-axis-parallel partitions in its isolation mechanism. Hence, iNNE does not have the same issue.

To analyse this issue, we draw contour maps of anomaly score in two dimensional space. We expect that an ideal anomaly detector should have tight contours separating regions which contain normal instances from the rest of space. Figure 5a shows a spiral shape dataset, and it has six anomalies in-between the spiral lines. Note that, these anomalies would be masked by normal instances when projected onto either of the two dimensions.


(a)

(b)

(c)

Figure 5: (a) Spiral dataset with 4000 normal instances (blue cross) and 6 anomaly instances (red diamond) (b) Contour graph of iNNE (t= 100, ψ= 256) anomaly score for spiral dataset. AUC = 1.00, Anomaly Ranking: 1 - 6. (c) Contour graph of iForest (t= 100, ψ= 256) anomaly score for spiral dataset. AUC = 0.86, Anomaly Ranking: 75, 320, 345, 354, 563, 1802

Figures 5b and 5c show the contour maps drawn by anomaly scores of iNNE and iForest, respectively.

The contour map of iNNE model data distribution well, and it also ranks the anomalies on top of the ranked list with AUC equals to 1.0. However, iForest produces jagged contours; and it gives low anomaly scores to the space in-between the spiral lines and places anomalies not at the top of the ranked list.

This result clearly highlights the issue iForest has with regard to such situations. However, the isolation mechanism of iNNE is able to overcome this weakness.

Scale-up test

When it comes to large datasets, execution time is a key factor of concern. Time complexity of a method is the deciding factor for its execution time. Most of the distance and density based anomaly detectors have a quadratic time complexity $O(n^2)$ due to nearest neighbour calculations, which can be reduce to $O(nlog(n))$ using some indexing method. Despite the fact that iNNE is a nearest neighbour method, it has a linear time complexity and can scale up to very large datasets.

An experiment was conducted to examine the increase in run time with increasing data size. We used the Mulcross data generator to generate 5 dimensional datasets with increasing data sizes. The generated data sizes are: 1000, 5000, 10000, 50000, 100000, 200000, 500000, 1000000, 5000000 and 10000000. Parameter k of LOF and ORCA is set to 50, which is a moderate value for this data set with clustered anomalies. The default settings of iForest were used: $t = 100$ and $\psi = 256$ [8]. iNNE used the following settings: $t = 100$ and $\psi = 32$.

Because LOF's memory requirement is high, LOF was executed with $64GB$ memory. iNNE, iForest and ORCA were executed with $32GB$ memory. LOF with R*-Tree indexing (LOFIndexed) and without indexing scheme (LOF) were conducted to examine the effect of indexing scheme.

We run each job up to a maximum of 20 days. With this time limit, LOF could only complete the task up to half a million instances; and ORCA could only complete the task up to a million instances.

Figure 6 shows the scale-up test result using 1000 instances as the base for the ratio calculations. The result shows that LOF and ORCA took significantly longer than iNNE and iForest, especially in large data sets. LOFIndexed has similar run time ratio as those of iNNE and iForest for data size of 1 million or less. However, LOFIndexed had a much steeper run time ratio beyond 1 million instances. It is apparent that LOF would be prohibitively expensive for data sets with 10 million instances which has a projected run time of 220 days. ORCA ran faster than LOF; but, it is still going to take a projected run time of 15 days for the data set with 10 million instances. Indexing has made LOF run faster; however, it still took more than 7 hours, compared with less than 2 hours by iNNE for the same data set. iForest is by far the most efficient of all these methods, with just 9 minutes execution time. Moreover, both iNNE and iForest have low gradients in the scale-up plot, which implies that the data size limit that they can handle is much higher than other distance-based methods. This experiment provides strong evidence that the ensemble approach of both iForest and iNNE is the key to handle large data sets.

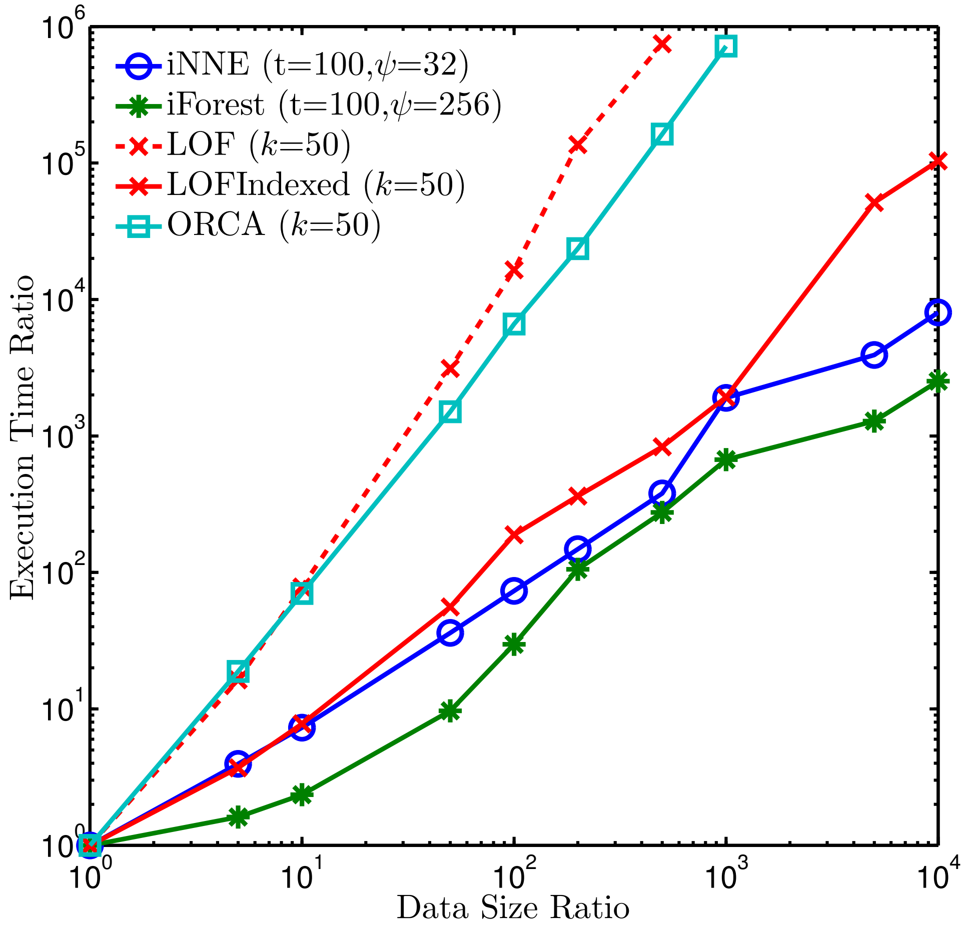


Figure 6: Scale-up test using Mulcross 5 dimensional datasets. Execution time for 10 *million* dataset iNNE: 1 hour 40 minutes, iForest: 9 minutes, LOF: 220 days (projected value), LOFIndexed: 7 hours 30 minutes, and ORCA: 15 days (projected value). Note that the starting overhead and the file I\O are excluded from time measurements.

3.3 Functions of unary and binary mass-based measures

In this project, we have found that the relative function of unary and binary mass-based measures cannot be easily distinguished because of the creation of relative mass which can be implemented as either unary or binary measure. Specifically, we have found that binary measures or relative mass are essential in addressing issues that are unable to be resolved using unary measures or mass. The issues that are being addressed are task specific. In this project, we have identified these issues in two tasks, i.e., anomaly detection and clustering.

Two outcomes prevail in anomaly detection:

- The earliest mass-based method that employs unary measure, iForest [8], has been identified to have difficulty in detecting local anomalies. The relative mass, implemented as a unary measure, is used to address this issue by simply replacing the measure used, i.e., path length which is a proxy to mass, to relative mass. Here, exactly the same trees are employed in both iForest and ReMass-iForest. Thus, ReMass-iForest has the new ability to detect local anomalies and has the same time complexity as iForest. This has been described in Section 2.2.1.

- In the previous project, we have converted LOF to DEMass-LOF by simply replacing the distance-based density estimator (using a binary measure) with mass-based density estimator (DEMass which uses a unary measure). In this case, DEMass-LOF runs orders of magnitude faster than LOF; and both have the equivalent detection accuracy, including the ability to detect local anomalies.

The above two relationships are depicted in Figure 7.
In clustering, the popular density clustering algorithm, DBSCAN (which uses a binary measure), has two key weaknesses: (a) it has high time and space complexities; and (b) it is unable to find all clusters of greatly varying densities. Unary mass measures are employed to address the first issue in two ways by replacing the distance-based density estimator with either mass estimator [12] or mass-based density estimator [9]. In this case, the unary measure approach only addresses the efficiency issue and both methods have the weakness as DBSCAN in finding all clusters of greatly varying densities. The relative mass, implemented as a binary measure in RMSCAN, is used to address the second issue [7]. The conversion from DBSCAN to RMSCSAN is simply replacing the distance measure with relative mass measure, leaving the rest of the algorithm unchanged. While this addressed the second issue, both RMSCAN and DBSCAN have the same time complexity.

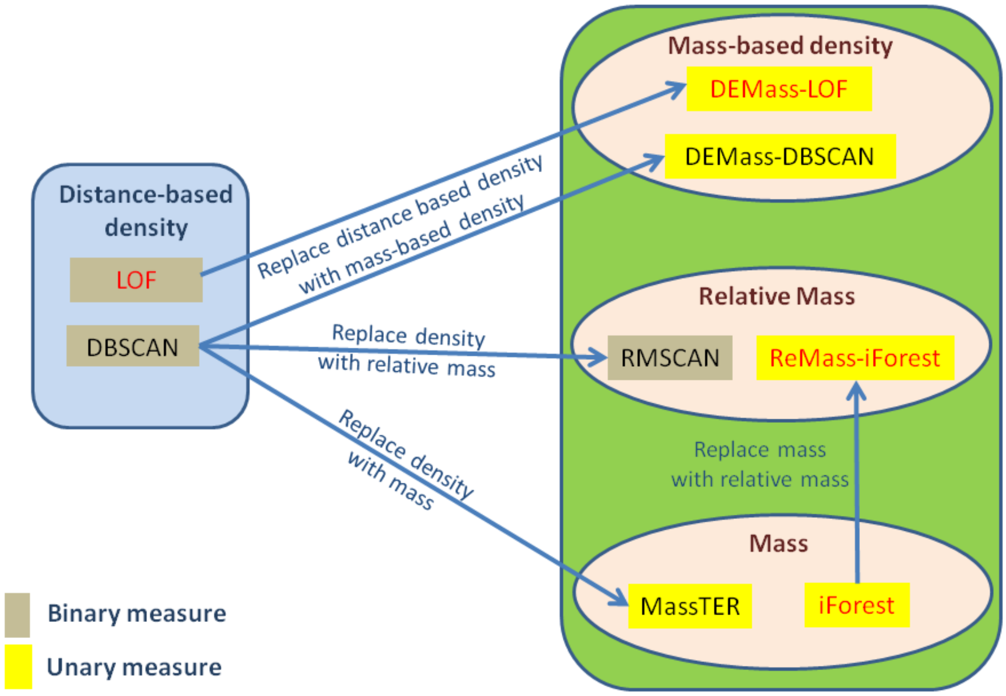


Figure 7: The conversion from distance-based density-based methods to mass-based methods in two tasks: anomaly detection (LOF, DEMass-LOF, iForest and ReMass-iForest) and clustering (DBSCAN, DeMass-DBSCAN, MassTER, RMSCAN.)

4 Final Remark

The two-year project has exceeded the planned objectives by investigating in four data mining tasks—two more than those specified in the project proposal. The project produced the first mass-based similarity measures and relative mass, and it has been successfully completed with the following outcomes:

1. Two non-metric similarity measures, `Massim` and m_p-dissimilarity, are proposed. Two implementations of `Massim` are created using balanced and imbalance trees. Preliminary assessments in classification, clustering and information retrieval tasks are very promising. The result of imbalanced tree, which is not presented in this report, can be found in [4]. m_p-dissimilarity has a simpler implementation without building trees or a model. This measure has been shown to perform better than

ℓ_p-norm and the cosine measure in kNN classification and information retrieval, especially in sparse high dimensional data sets.

2. Three implementations of relative mass are proposed since the introduction of mass estimation in 2010. The first two implementations of relative mass have been created using trees and nearest neighbour. The assessments in anomaly detection are very conclusive: relative mass is better than mass without any disadvantage. The third implementation has been applied to solve a long outstanding problem in density-based clustering algorithms, i.e., their inability to identify all clusters of hugely varying densities. There were many attempts to solve this problem; but these solutions were proposed without first identifying the exact conditions under which the density-based clustering algorithms will fail. In contrast, our solution is a principled approach targeted at the identified conditions.

Both of these results represent a significant milestone in mass estimation research. The non-metric similarity measures are a generalisation of mass estimation from a unary function to a binary function, enabling a similarity between two instances to be measured using a measure which is primarily relied on data distribution. This is in sharp contrast with distance-based measure which is based solely on positions in the feature space. Relative mass is an interesting research topic because it can be applied as a unary function or a binary function, depending on the task at hand: the application of relative mass in information retrieval and clustering tasks can be interpreted as a similarity measure, while it is a unary function in anomaly detection tasks.

5 List of Publications and Significant Collaborations that resulted from AOARD supported projects

5.1 List of peer-reviewed journal publications:

[1] Jonathan R. Wells, Kai Ming Ting and Takashi Washio. (2014) LiNearN: A New Approach to Nearest Neighbour Density Estimator. *Pattern Recognition.* Vol. 47, No. 8, 2702-2720. Elsevier.

[2] Sunil Aryal and Kai Ming Ting. (2015) A generic ensemble approach to estimate multi-dimensional likelihood in Bayesian classifier learning. *Computational Intelligence.* http://onlinelibrary.wiley.com/doi/10.1111/coin.12063/abstract

5.2 List of peer-reviewed conference publications

[3] Sunil Aryal, Kai Ming Ting, Gholamreza Haffari and Takashi Washio. (2014) mp-dissimilarity: A data dependent dissimilarity measure. *Proceedings of the 2014 IEEE International Conference on Data Mining.* 707-711.

[4] Sunil Aryal, Kai Ming Ting, Jonathan R. Wells and Takashi Washio. (2014) Improving iForest with Relative Mass. *Proceedings of the 18th Pacific-Asia Conference on Knowledge Discovery and Data Mining.* 510-521.

[5] Tharindu R. Bandaragoda, Kai Ming Ting, David Albrecht, Fei Tony Liu and Jonathan R. Wells. (2014) Efficient Anomaly Detection by Isolation Using Nearest Neighbour Ensemble. *Proceedings of the 2014 IEEE International Conference on Data Mining Workshop on Incremental Classification, Concept Drift and Novelty Detection.*

5.3 Papers currently submitted for review

[6] Kai Ming Ting, Thilak Laksiri Fernando and Geoffrey I. Webb. Mass-based Similarity Measure: An Effective Alternative to Distance-based Similarity Measures. Technical Report 276, 2013.

[7] Ye Zhu, Kai Ming Ting, Mark J. Carman and Thilak Laksiri Fernando. RMSCAN: A principal approach to recondition density-based clustering algorithms to successfully find clusters with varying densities. Submitted to *Data Mining and Knowledge Discovery Journal.*

5.4 Reference

[8] Fei Tony Liu, Kai Ming Ting, and Zhi-hua Zhou. (2008) Isolation Forest. *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining.* Washington, DC, USA: IEEE Computer Society, 413–422.

[9] Kai Ming Ting, Takashi Washio, Jonathan R. Wells, Fei Tony Liu and Sunil Aryal. (2013) DEMass: a new density estimator for big data. *Knowledge and Information Systems.* 35(3) 493–524.

[10] Sunil Aryal and Kai Ming Ting (2013) MassBayes: A new generative classifier with multi-dimensional likelihood estimation. *In Proceedings of the 17th Pacific-Asia Conference on Knowledge Discovery and Data Mining.* Lecture Notes in Computer Science. Springer Berlin Heidelberg. 136-148

[11] Kai Ming Ting, Guang-Tong Zhou, Fei Tony Liu and Tan Swee Chuan (2013) Mass Estimation. *Machine Learning Journal.* Vol. 90, Issue. 1, 127-60.

[12] Kai Ming Ting and Jonathan R. Wells. (2010) Multi-dimensional mass estimation and mass-based clustering. *Proceedings of IEEE International Conference on Data Mining.* 511520.

[13] A.Y. Ng, M. Jordan, and Y. Weiss. (2001) On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems.* 14, 849–856.

[14] J. Liu, and J. Han. (2014) Spectral Clustering. Chapter 8 in *Data Clustering.* Editors: C.C. Aggarwal and C.K. Reddy. CRC Press.

[15] M.M. Breunig, Hans-Peter Kriegel, R.T. Ng and Jörg Sander. (2000) LOF: Identifying Density-Based Local Outliers. *Proceedings of ACM SIGMOD International Conference on Management of Data.* 93–104.

[16] Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu. (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining.* 226–231.

5.5 Significant collaborations

I attended and made a presentation at the program review in the area of Computational Cognition and Robust Decision Making, at AFOSR headquarter in Arlington Virginia on 9-13 December 2013. I had a meeting with Hiroshi Motoda annually during the project period and made a presentation in each meeting.

Takashi Washio of Osaka University has contributed significantly in the project, resulting in three joint papers [1, 3, 4]. Monash colleagues, Geoffrey Webb, Gholamreza Haffari, David Albrecht and Mark Carman, have collaborated in this project, and they are the co-authors in five papers.

Note

Mass-based similarity papers:

- Paper [6] provides the theory and assessments of the first version of mass-based similarity measure and a tree implementation.
- Paper [3] presents the a simplified version of mass-based similarity without building trees or a model.

Relative Mass papers:

- Paper [4] reveals the first relative mass implementation using tree and assess its performance in anomaly detection and information retrieval.
- Paper [5] proposes the first implementation of relative mass using nearest neighbour approach, using a variant described in [1].
- Paper [7] presents the first relative mass similarity measure and uses it to replace density measure in DBSCAN to overcome the one key weakness of density-based clustering algorithms.

Papers produced as a result of previous AOARD projects:

- Paper [1] presents the first linear time complexity nearest neighbour algorithm. This work was supported by a previous AOARD project.
- Paper [2] extends the work previously published in PAKDD-2013 [10] to present the first generic approach to estimate multi-dimensional likelihood $p(\mathbf{x}|y)$ directly by aggregating $p_i(\mathbf{x}|y)$ estimated from an ensemble of estimators where each estimator is constructed from a small fixed-size random sub-sample of data $\mathcal{D}_i \subset D$ $(i = 1, 2, ..., t)$. This is a generic approach because $p_i(\mathbf{x}|y)$ can be estimated using different data modelling methods. DEMass-Bayes [9] and MassBayes [10] are two realisations of the proposed generic approach. In this paper, we introduce an additional realisation of the proposed generic approach called ENNBayes along with MassBayes. ENNBayes estimates $p_i(\mathbf{x}|y)$ from $\mathcal{D}_i$ using a nearest neighbour density estimator which is a variant described in [1].

Wikipedia entry

The Wikipedia entry of mass estimation has been established in March 2014. It can be found at http://en.wikipedia.org/wiki/Mass_estimation.

Software Downloads

The source codes of multi-dimensional mass estimation, DEMass-DBSCAN, DEMass-Bayes and MassBayes, algorithms proposed in papers [11, 9, 10], are made available at `http://sourceforge.net/projects/mass-estimation/`